

Intelligent Decision Support for Medication Review

Ivan Bindoff¹, Peter Tenni², Dr. Byeong Ho Kang¹, Prof. Gregory Peterson²

¹University of Tasmania, School of Computing
{ibindoff, bhkang}@utas.edu.au

²University of Tasmania, Unit for Medical Outcomes and Research Evaluations
{Peter.Tenni, G.Peterson}@utas.edu.au

Abstract. This paper examines an implementation of a Multiple Classification Ripple Down Rules system which can be used to provide quality Decision Support Services to pharmacists practicing medication reviews (MRs), particularly for high risk patients. The system was trained on 84 genuine cases by an expert in the field; over the course of 15 hours the system had learned 197 rules and was considered to encompass around 60% of the domain. Furthermore, the system was found able to improve the quality and consistency of the medication review reports produced, as it was shown that there was a high incidence of missed classifications under normal conditions, which were repaired by the system automatically.

1 Introduction

Sub-optimal drug usage is a serious concern both in Australia and overseas [1, 2], resulting in at least 80,000 hospital admissions annually - approximately 12% of all medical admissions and reflecting a cost of about \$400 million annually, with the majority of these affecting elderly patients [3]. MRs are seen as an effective way to improve drug usage. However, the quality of MRs produced is inconsistent across reviewers. Further to this, many community-based pharmacists are still unwilling to undertake this new role, citing reasons including fear of error and a lack of confidence [4].

This paper proposes a different approach to improving the quality of the MRs, and possibly even improving the uptake of the role within the pharmaceutical community. It is suggested that the answer may lie in the development of medication management software which includes Intelligent Decision Support features. To date, the majority of incarnations of medication management software for producing MRs has lacked any form of genuine Decision Support features [5]. Unfortunately, Knowledge Based System (KBS) techniques which may be suitable to this problem have been designed to handle steadfast, well defined sets of knowledge, and have historically not been well suited to poorly structured or dynamic sets of knowledge such as the set found in the domain of MR. However, newer techniques such as Case Based Reasoning (CBR) and Ripple Down Rules (RDR) may offer new possibilities in handling knowledge of this kind, since they are easily, even naturally, maintainable and alterable [6, 7].

2 Medication Reviews

MR is a burgeoning area in Australia and other countries, with MRs seen to be an effective way of improving drug usage and reducing drug related hospital admissions, particularly in the elderly and other high risk patients [1, 3]. This has prompted the Australian government to initiate the Home Medicines Review scheme (HMR) and the Residential Medication Management Reviews (RMMRs) scheme. These schemes provide remuneration to pharmacists performing MRs via a nationally funded program [3]. However, it is known that despite Residential Medication Management Reviews (RMMRs) being introduced in 1997 they still do not have a conceptual model for delivery, which has resulted in a wide range of differing qualities of service being provided [4].

To perform a MR, Pharmacists assess potential Drug Related Problems (DRPs) and Adverse Drug Events[†] (ADEs) in a patient by examining various patient records, primarily their medical history, any available pathology results, and their drug regime (past and current) [8]. The expert looks for a variety of indicators between the case details provided checking for known problems, such as an: Untreated Indication – where a patient has a medical condition which requires treatment but doesn’t have the treatment; Contributing Drugs – where a patient has a condition and is on a drug which can cause or exacerbate said condition; High Dosage – where a patient is potentially on a too high dosage because of a combination of drugs with similar ingredients; Inappropriate Drug – where a patient is on a drug that is designed to treat a condition they don’t seem to have or is contraindicated in their condition; and many others besides. Once these indicators have been identified a statement is produced explaining each problem, or potential problem, and often what the appropriate course of action is.

3 Methodology

In order to produce a medication management system with intelligent decision support features it was necessary to produce two major software elements. The first was a standard implementation of a database “front-end” from which it is possible for a user to enter all the details of a given patient’s case, or at least those parts which are relevant to the chosen domain. The second was an implementation of a Multiple Classification Ripple Down Rules engine which can sufficiently encapsulate the types of conditions and knowledge required for the domain and facilitate the design of an interface from which the engine can be operated, particularly during the Knowledge Acquisition phase.

[†] defined by the World Health Organisation as being “an injury resulting from medical intervention related to a drug.” 2. Bates, D., et al., *Incidence of adverse drug events and potential adverse drug events. Implications for prevention. ADE Prevention Study Group.* JAMA, 1995(274): p. 29-34.

3.1 Database

The design of the database to store the MR cases was relatively trivial, and will not be given much detail here. The preliminary design idea was taken from existing medication management software packages, and then extensively modified to allow for proper computerized analysis. The 126 cases considered in this study were then inserted into the database using a simple script which converted them from their current Mediflags [9] format.

3.2 Ripple Down Rules

Ripple Down Rules (RDR) is an approach to building KBSs that allows the user to incrementally build the knowledge base while the system is in use, with no outside assistance or training from a knowledge engineer [7]. It generally follows a forward-chaining rule-based approach to building a KBS. However, it differs from standard rule based systems since new rules are added in the context in which they are suggested.

Observations from attempts at expert system maintenance lead to the realisation that the expert often provides justification for why their conclusion is correct, rather than providing the reasoning process they undertook to reach this conclusion. That is, they say ‘why’ a conclusion is right, rather than ‘how’. An example of this would be the expert stating “I know case A has conclusion X because they exhibit features 1, 4 and 7”. Furthermore, experts are seen to be particularly good at providing comparison between two cases and distinguishing the features which are relevant to their different classifications [10]. With these observations in mind an attempt was made at producing a system which mimicked this approach to reasoning, with RDR being the end result.

3.3 Structure

The resultant RDR structure is that of a binary tree or a decision list [11], with exceptions for rules which are further decision lists. The decision list model is more intuitive since, in practice, the tree would have a fairly shallow depth of correction [12]. The inferencing process works by evaluating each rule in the first list in turn until a rule is satisfied, then evaluating each rule of the decision list returned by that satisfied rule similarly until no further rules are satisfied. The classification that was bound to the last rule that was satisfied is given.

RDR can be viewed as an enhancement to CBR [6, 13, 14], with RDR providing a utility, in the form of an algorithm, a structure and rules, with which to demonstrate which parts of the case are significant to a particular classification [15].

3.4 Multiple Classification Ripple Down Rules

The RDR method described above is limited by its inability to produce multiple conclusions for a case. To allow for this capability - as this domain must - MCRDR should be considered [16] to avoid the exponential growth of the knowledge base that would result were compound classifications to be used.

MCRDR is extremely similar to RDR, preserving the advantages and essential strategy of RDR, but augmented with the power to return multiple classifications. Contrasting with RDR, MCRDR evaluates all rules in the first level of the knowledge base then evaluates the next level for all rules that were satisfied and so on, maintaining a list of classifications that should fire, until there are no more children to evaluate or none of the rules can be satisfied by the current case [12]. An example of this can be seen in **Fig. 1**.

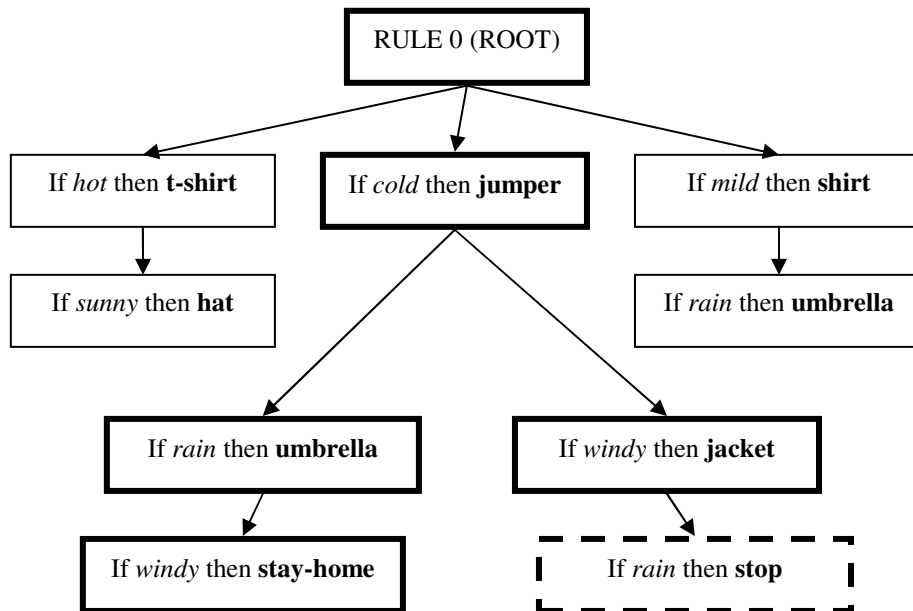


Fig. 1. The highlighted boxes represent rules that are satisfied for the case (cold, rain, windy), the dashed box is a potential stopping rule the expert may wish to add [17]

3.4 Knowledge Acquisition

Knowledge Acquisition is required when a case has been classified incorrectly or is missing a classification. It is divided into three separate steps: Acquiring New Classification (or Conclusion), Locating the New Rule, and Acquiring the New Rule. It should be noted that the order of applying steps one and two is unimportant to the

validity of the method [12], and that for the purposes of this experiment it made sense to locate the rule before acquiring the new classification. Hence this is what was done.

Acquiring the New Classification is trivial; the system merely prompts the expert to state it [12]. To Acquire the New Rules the expert is asked to first select valid conditions from the current case that indicate a given classification. The rule they have created thus far is then compared against the cornerstone case base. If any cornerstone cases would fire on this new rule the expert is asked to select from a difference list (see **Table 1**) between the presented case and one of the cornerstone cases. A cornerstone case is a case for which the knowledge had previously been modified and which is valid under the current context [18]. The system then re-tests all cornerstone cases in the list against the appended set of conditions, removing cases from the list that are no longer satisfied. The system repeats this process until there are no remaining cornerstone cases in the list to satisfy the rule [12] or alternatively the expert has stated explicitly that the cornerstone cases that remain *should* fire on the new rule and this new classification was simply missed when the cornerstone case was originally considered.

Table 1. Example of a decision list from [7, 15, 17, 19]. The list can contain negated conditions

<i>Cornerstone case</i>	<i>Current test case</i>	<i>Difference list</i>
Rain	Rain, Meeting	Meeting
Meeting	Meeting	Not applicable
Hot		Not(Hot)

To determine where the new rule must go it must first be determined what type of wrong classification is being made. The three possibilities are listed in **Table 2**.

Table 2. The three ways in which new rules correct knowledge base [12]

Wrong Classifications	To correct the Knowledge Base
Wrong classification to be stopped	Add a rule (stopping rule) at the end of the path to prevent the classification
Wrong classification replaced by new classification	Add a rule at the end of the path to give the new classification
A new independent classification	Add a rule at a higher level (to the root) to give the new classification

4 Results and Discussion

The system was handed over to the expert with absolutely no knowledge or conclusions pre-loaded. The expert was wholly responsible for populating the knowledge base. Over the course of approximately 15 hours they were able to add the rules re-

quired to correctly classify 84 genuine MR cases that had been pre-loaded into the system.

4.1 Growth of Knowledge Base

It is observed in **Fig. 2** that the number of rules in the system progressed linearly as more cases were analysed, at a reasonably consistent rate of about 2.3 rules per case. This suggests that the system was still in a heavy learning phase when the experiment was finished, since it has previously been observed that RDR systems will show a flattening pattern in the rate of growth of the knowledge base at approximately 80% of domain coverage [12]. This has complications for many of the remaining tests, in that their results must be understood to reflect the knowledge base while it is still learning heavily. The general conclusion that can be applied here is that most results will be expected to improve with additional testing, and that further testing is indeed required. This is because without demonstrating that the rate of learning has begun to slow down it is impossible to adequately prove that the heavy learning phase, which requires a significantly higher level of expert maintenance, will cease.

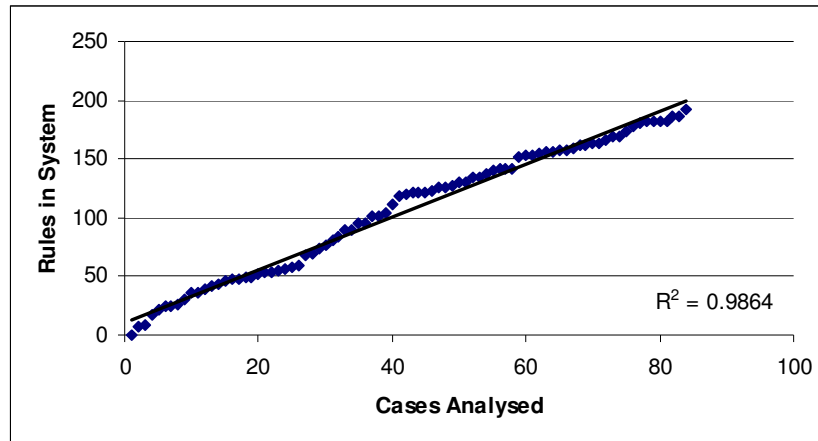


Fig. 2. The number of rules in the system grows linearly as more cases are analysed

4.2 Correct Conclusions Found

It was estimated by the expert at the time of cessation of the experiment that the system had encapsulated around 60% of the domain [20], this estimation is supported by the evidence shown in **Figure 3**. It can be seen that the average number of correct classifications the system provided rose quite steadily into the 60th percentile, although the percentage correct from case to case did vary quite a lot, as is to be expected when the system is still in the heavy learning phase.

The expert predicted potential classification rates in the order of 90% [20], so considering 84 cases had been analysed it could be estimated that in order to reach this

rate at least another 40 cases should be analysed, and it would be unexpected if the number of additional cases needed to be analysed exceeded about 120, based on previous figures found for systems of this kind [12, 18]. These figures are justified by following the trend-line in Figure 3 which shows the clustered average of correct conclusions provided by the system for each group of 5 cases analysed, although it is conceded that this trend-line is only a rough approximation. If it is followed linearly as demonstrated thus far it reaches 90% at approximately 120 cases, if it is assumed that this trend-line may begin to plateau though, as expected, it is possible that the number of extra cases required may grow considerably, to reflect the slower rate of learning.

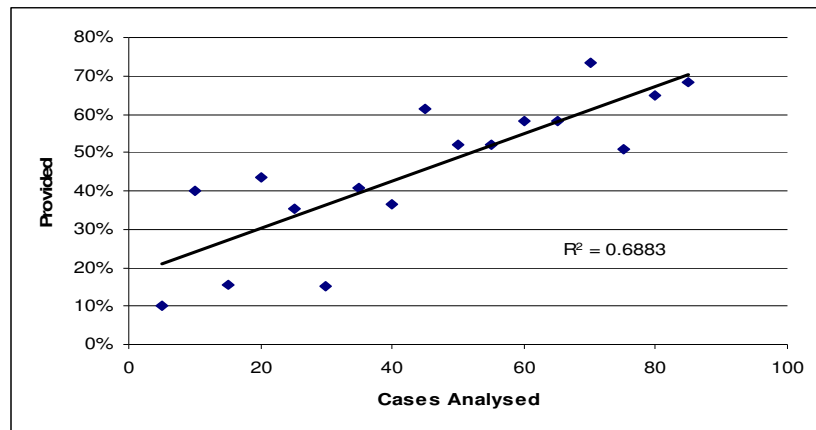


Fig. 3. The percentage of conclusions provided by the system for each case that were already correct

4.3 Classifications Found, Expert vs. System

The results shown in Figure 4 are very convincing, with the system sometimes finding half again as many classifications per case as the expert and quite consistently remaining at least one classification ahead. It should be re-iterated that the system found all these classifications using only a smaller set of the same knowledge the expert had. This suggests the expert consistently misses classifications they should find. In other words, they just don't notice them on the particular case. The system does not suffer from this, it will notice anything that it is trained to know about without exception.

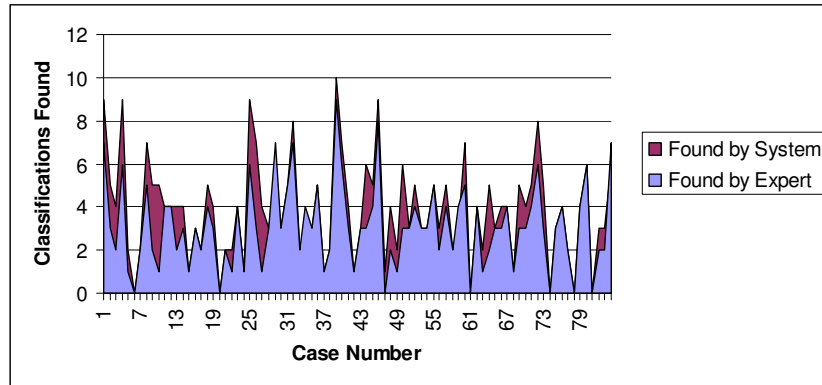


Fig. 4. The system found significantly more correct classifications than the expert

4.4 Percentage of Classifications Missed

It was found that the expert often appended classifications to previous cases after the systems prompting, particularly early in the systems training. Evidence of this is shown in **Fig. 5**. The percentage reduced dramatically even after only a small number of cases, suggesting the system was rapidly helping to reduce the experts rate of missed classifications, by suggesting the classifications for them, rather than making the expert notice themselves. The trend-line in **Fig. 5** is only an approximation, since relatively few cases have been analysed thus far, and noise is still significant.

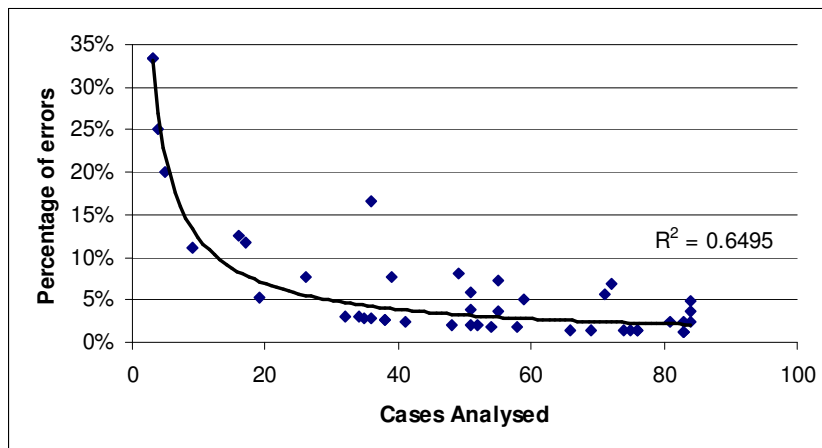


Fig. 5. The percentage of cases that gained new classifications

4.5 Total Errors per Case

It was found that the rate of error in each case was quite high, averaging 13.4% and with some going over 50%. Clearly the expert is making errors regularly, as was expected, and yet these numbers would be expected to be even higher were more complete training done. It is important to note that the results shown in **Fig. 6** are representative of all the errors (missed classifications) that the system has fixed through the normal course of operation, and not the actual number of errors per case. This figure suggests that over 1 in 9 classifications are missed, although it is unclear what type of classifications these are. What level of threat are these classifications likely to pose? One would like to assume that the expert would not miss life threatening classifications, because they would have a particular focus on these, but additional experimentation is clearly required to determine what kinds of classifications the expert is missing and what the consequences of this is.

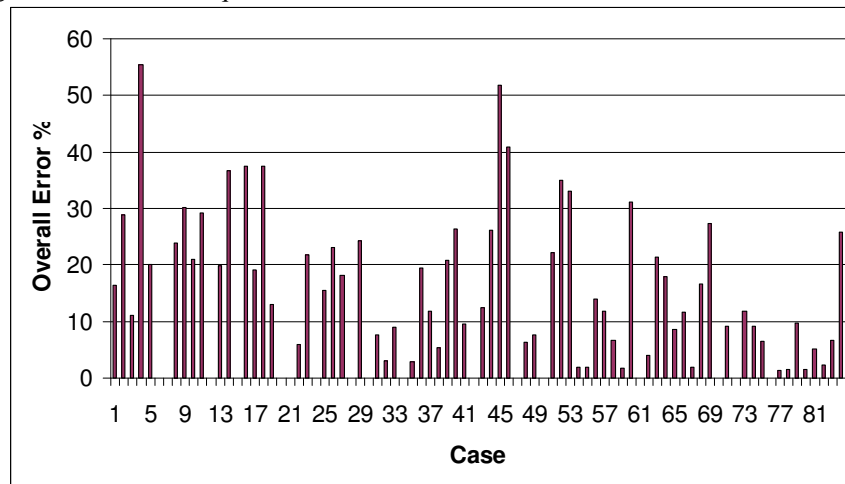


Fig. 6. The final percentage of classifications missed by expert per case

4.6 Maintainability and Usability

It was considered possible that the domain of MR might damage the maintainability and usability of the system due to both its inconsistent/dynamic nature, and the large number of variables within each case. It was considered possible that the dynamic nature of the domain might result in a need for an excessive number of exceptions to be added to the knowledge base, and that the large number of variables within each case may have resulted in an excessive number of conditions required for each rule. Each of these afflictions would increase the time taken to maintain/use the system and possibly make it untenable. As such, tests were carried out to determine whether these figures deviated remarkably from the normal range to be expected in a system of this nature.

Time Spent: Adding Rules

Previous RDR systems have reported figures of around 3 minutes per rule [18], and it was found that this system continued that trend, with the average time taken to add a rule being 183 seconds (3 minutes).

Time Spent: Analysing Cases

It was found that the average time taken for the expert to complete a case analysis was about 10 minutes (621 seconds). This average extended over the entire 84 cases gives a total expert time taken as about 15 hours as reported earlier. Some cases were done in as few as 2 minutes when no or few new classifications were required, although the process did sometimes reach over 20 minutes.

Cornerstones Seen

The results here are promising from a usability point of view, with the expert rarely having to consider cornerstone cases in the creation of rules, with the majority of rules having no cornerstone cases to consider. In fact the expert saw an average of only 0.42 cornerstones per rule. What this means is that the expert should be able to add rules relatively quickly, with the time required to validate their rules being small.

4.7 Structure of the Knowledge Base

It can be determined from

Table 3 that the structure of the knowledge base tree was extremely shallow and branchy, meaning the possibility of an excessive number of exceptions has not, at least at this point, come to light at all.

Table 3. Structure of the Knowledge Base Tree

Tree Property	Value
Average Depth	1.30
Depth 1	139
Depth 2	53
Depth 3	3

The nature of the rules in the knowledge base is also of interest, with further support for the maintainability of the system shown in the fact that the average number of conditions selected in a rule was only 1.7, with longer rules of 4 or 5 conditions being virtually non-existent and no rules with 6 or more being present.

To get a more complete view of the knowledge base it is necessary to analyse what outputs the rules map to. With the knowledge base that was built in the process of this experiment 85 individual conclusions were defined. When it is considered that every rule except stopping rules, of which there are 154, is linked to a conclusion it can be seen that there is 1 conclusion for every 1.8 rules, as can be demonstrated with the data used in **Fig. 7**. It is evident from this figure that, although most conclusions are

only used by one rule, some conclusions are used very often. In other words they have many different sets of conditions which can lead to them.

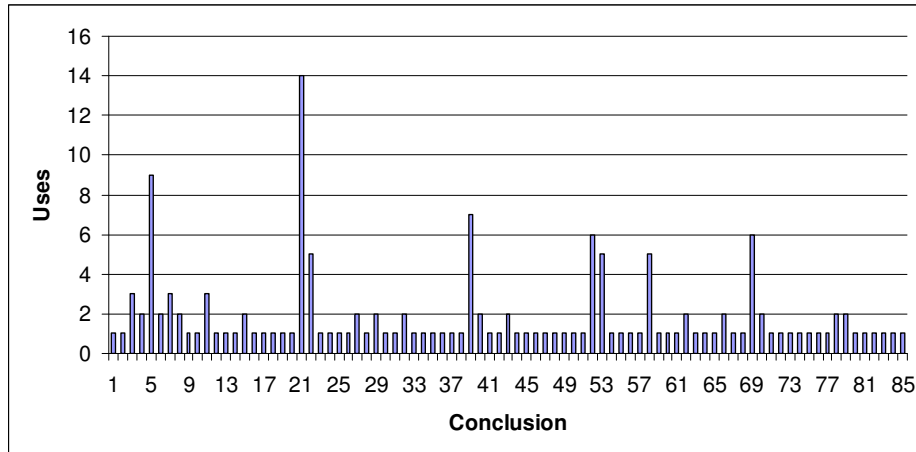


Fig. 7. How many times each conclusion was used

5 Conclusions

Initial experimentation suggests that the proposed method using MCRDR can successfully represent knowledge where the knowledge sources (human experts) are inconsistent. The system is shown to have reached about an 60% classification rate with less than 15 expert hours and only 84 cases classified – a good outcome in the circumstances. The knowledge base structure does not show any major deviations from what would be anticipated in a normal MCRDR system at this stage. The maintainability of the system does not appear to have been adversely affected thus far, with the expert being faced with only few cornerstone cases during the knowledge base validation, and the time taken to add rules being negligible.

From a MR perspective the system is seen to be capable of: providing classifications for a wide range of Drug Related Problems; learning a large portion of the domain of MRs quickly; producing classifications in a timely manner; and importantly, vastly reducing the amount of missed classifications that would otherwise be expected of the reviewer. It is expected that this system, or a future incarnation of this system, would be capable of achieving classification rates around 90% [20]. If this figure is to be realised it is possible that this system would be capable of achieving three major goals:

- Reducing the amount of missed classifications
 - Thus improving the consistency (quality) of service
- Improving the confidence of potential medication reviewers

It has already been noted that the number of errors this system detected and repaired was significant, and the number of errors was seen to reduce as the expert populated the knowledge base and this result alone would be enough to warrant further work. It has also been observed that the amount of time taken to perform a MR using this system should not be adversely affected. As for the final point it is anticipated that a system such as this might improve reviewer's confidence by providing a reliable second level of checking for their conclusions, since this system is designed and trained to act as an expert in the field did.

6 Further Work

It should be noted that the system built for this study was intended only for an initial proof of concept testing. Further testing is needed over a broader range of cases to verify the results shown in this paper, however initial testing does not suggest any insurmountable problems will arise. On top of this, the system could be more powerful and better encompass the domain by including the additional features mentioned below.

6.1 Time Series Data

An important feature that was missing from the prototype was the handling of time series data, such that the expert would be able to define rules such as "increasing" or "decreasing" for things like Weight, Blood Pressure, or a Pathology result. Further still, they might define things like "recent" or "old", which check whether a result is older or younger than defined thresholds, newest, oldest, average and others. As the system stands it will fire on a rule that states "Creatinine > 0.12" even if the result which says their Creatinine level was 0.13 was taken 15 years prior. This is undesirable, with the meaning of the results varying across periods of time such that the expert may wish to define rules based on different types of results.

6.2 Standardisation

It was observed that the knowledge acquisition workload is increased when inconsistent nomenclature is allowed, such as it so often is in many medical systems. To prevent this increased workload for the expert, it would be prudent to derive and enforce a strict scheme for the data input. A possible complication is that users may find it difficult to locate options which are not named as expected. To handle this it would be possible to implement another interpretive layer of hierarchy, essentially allowing the user to use their own preferred nomenclature, and then defining within the system that their chosen nomenclature is synonymous to whichever standardised equivalent is selected by the system designers.

References

1. Peterson, G., *Continuing evidence of inappropriate medication usage in the elderly*, in *Australian Pharmacist*. 2004. p. 2.
2. Bates, D., et al., *Incidence of adverse drug events and potential adverse drug events. Implications for prevention*. ADE Prevention Study Group. JAMA, 1995(274): p. 29-34.
3. Peterson, G., *The future is now: the importance of medication review*, in *Australian Pharmacist*. 2002. p. 268-75.
4. Rigby, D., *The challenge of change - establishing an HMR service in the pharmacy*, in *Australian Pharmacist*. 2004. p. 214-217.
5. Kinrade, W., *Review of Domiciliary Medication Management Review Software*. 2003, Pharmacy Guild of Australia. p. 77.
6. Aamodt, A. and E. Plaza, *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*, in *AICom - Artificial Intelligence Communications*. 1994. p. 39-59.
7. Compton, P., et al. *Knowledge Acquisition without Analysis*. in *Knowledge Acquisition for Knowledge-Based Systems*. 1993. Springer Verlag.
8. Tenni, P., et al. to I. Bindoff. 2005.
9. Bonner, C., *MediFlags*. 2005.
10. Compton, P. and R. Jansen. *A philosophical basis for knowledge acquisition*. in *European Knowledge Acquisition for Knowledge-Based Systems*. 1989. Paris.
11. Rivest, R., *Learning Decision Lists*, in *Machine Learning*. 1987. p. 229-246.
12. Kang, B., P. Compton, and P. Preston, *Multiple Classification Ripple Down Rules*. 1994.
13. Kolodner, J., R. Simpson, and K. Sycara-Cyranski. *A Process Model of Cased-based Reasoning in Problem Solving*. in *International Joint Conference on Artificial Intelligence*. 1985. Los Angeles: Morgan Kaufmann.
14. Kolodner, J.L., *Special Issue on Case-Based Reasoning - Introduction*. Machine Learning, 1993. **10**(3): p. 195-199.
15. Kang, B. and P. Compton, *A Maintenance Approach to Case Based Reasoning*. 1994.
16. Kang, B., P. Compton, and P. Preston. *Multiple Classification Ripple Down Rules: Evaluation and Possibilities*. in *AIII-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems*. 1995. Banff.
17. Bindoff, I., *An Intelligent Decision Support System for Medication Review*, in *Computing*. 2005, University of Tasmania: Hobart. p. 65.
18. Preston, P., G. Edwards, and P. Compton. *A 2000 Rule Expert System Without a Knowledge Engineer*. in *AIII-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems*. 1994. Banff.
19. Compton, P. and R. Jansen. *Cognitive aspects of knowledge acquisition*. in *AAAI Spring Consortium*. 1992. Stanford.
20. Tenni, P. to I. Bindoff. 2005.